EXHIBIT 13

A Sample Session

1. The user launches the remote app on their device.
    - The remote tells the server a user has turned on their remote by registering a session. The remote also hangs a get request to listen for incoming messages from the server.
    - The server replies with loungeScreenDisconnected message, since we don't have a screen yet.
2. The user opens Leanback.
    - Leanback registers with the server in a similar manner to the remote. It also hangs a get request that awaits incoming remote commands.
    - The server replies with remoteControlConnected message since we have a remote.
3. The server sends a loungeScreenConnected message to the remote.
4. The remote sends the current playlist to the screen
5. The user closes the Leanback player.
    - Leanback sends a loungeScreenDisconnected to the server with now playing info; if it's the last screen in the session the server forwards it to the remote
6. Remote starts playing the video locally

Server to Remote messages

loungeScreenConnected()

The server informs the remote that there is at least one screen connected in the session.

loungeScreenDisconnected(feed_name, type, title, video_id, current_time)

The server informs the remote that there is no screen connected in the session. Optionally, if there was a screen before, the server sends info about what the screen was playing when it was disconnected.

- video_id - the encrypted video id of the currently playing video
- feed_name - title of the feed
- type - type of the feed
- current_time - playback position in the video

Server to Screen messages

remoteConnected()

The server informs the screen that there is a remote connected.

remoteDisconnected()

The server informs the screen that the remote was disconnected.